

ملاحظة :

في الحلقة السابقة في تمرين while كتب برنامجنا :  
while (i < n);

سلسلة مصفوفات :

المصفوفة ذات البعد الواحد : مصفوفة  
[عندئذ] اسم المصفوفة ب نوع المتغيرات

int A[5];

A[0], A[1], A[2], A[3], A[4]

مثال :

للتوصل إلى عنوان المصفوفة نكتب [دليل]

في خلال عام وطباعة عناصر المصفوفة نكتب

من خلال المصفوفات علم تحديد عدد عناصر المصفوفة :

int A[n];

مثال :

وعناصر المصفوفة تكون متتالية وكل عنصر بحجم 2 byte



A[0] A[1] A[2] A[3] A[4]

تمرين [1] :

باستخدام المصفوفات اكتب برنامج يقوم بما يلي :

1- ادخال عناصر المصفوفة ذات بعد واحد

2- حساب مجموع عناصر المصفوفة :

3- ترتيب عناصر المصفوفة بشكل صحيح

تمرين [2] :

اكتب برنامج يسمح بإدخال وطباعة عناصر المصفوفة :

```
#include <iostream.h>
```

```
#define n 5
```

```
void main()
```

```
{ int A[n], i;
```

```
// ادخال عناصر المصفوفة :
```

```
for (i = 0; i < n; ++i)
```



```

cin >> A[i];
for (i=0; i<n; ++i)
    cout << A[i];
}

```

لنوضح أكثر:

```

for (i=0; i<n; ++i;
    cout << "In A[" << i << "]" = ";
    cin >> A[i];
}
A[0] = 5
A[1] = 8

```

هذا المربع 1

```

#include <iostream.h>

```

```

#define n 5

```

```

void main()

```

```

{

```

```

    int A[n], B[n], k, s;

```

```

    for (i=0; i<n; ++i)

```

```

        cin >> A[i];

```

```

        s=0;

```

```

        for (i=0; i<n; ++i)

```

```

            s = s + A[i]; cout << "s = " << s;

```

```

            cout << "k = "; cin >> k;

```

```

for (i=0; i<n; ++i) for (i=0; i<n; ++i)

```

```

{

```

```

    B[i] = k * A[i];

```

```

    cout << "In B[" << i << "]" = " << B[i];

```

```

}

```

```

}

```



ولا حلقه :

يكتب على الواقع تعريف مصفوفة واعطاد لها قيم ابتدائية مما سمى به حلقة For مثال :

```
int A[3] = {8, 7, 13};
```

نفس المصفوفة ذات البعدين :  
 [ دليل ثاني ] [ دليل اول ] اسم المصفوفة - نوع متغيرات  
 مثال :

```
int A[3][4];
```

عدد الأعمدة ← عدد الأسطر

هنا لدينا مصفوفة مكونة من (3) أسطر و (4) أعمدة .  
 الدليل الأول يمكن عد الأسطر ولعل الثاني يمكن عد الأعمدة .  
 عناصر المصفوفة :

A[0][0]	A[0][1]	A[0][2]	A[0][3]
A[1][0]	A[1][1]	A[1][2]	A[1][3]
A[2][0]	A[2][1]	A[2][2]	A[2][3]

تربيع :

- ١- برنامج مصفوفة ذات البعدين يكتب برنامج مصفوفة مربع
- ٢- ادخال عناصر مصفوفة ذات بعدين
- ٣- طباعة هذه العناصر
- ٤- طباعة منقول المصفوفة (استبدال الأسطر بأعمدة الأسطر)
- ٥- طباعة عناصر القطر الرئيسي وعدد

الحل :

```
1- #include <iostream.h>
   #define n 3
   void main()
   {
       int A[n][n], i, j, k = 0;
       عدد عناصر القطر الرئيسي
```



```
for (i=0; i<n; ++i)
```

```
for (j=0; j<n; ++j)
```

```
cin >> A[i][j];
```

إدخال

2)-

```
for (i=0; i<n; ++i)
```

```
for (j=0; j<n; ++j)
```

```
cout << A[i][j];
```

3)-

```
for (i=0; i<n; ++i)
```

```
for (j=0; j<n; ++j)
```

```
cout << A[j][i];
```

4)-

```
for (i=0; i<n; ++i)
```

```
for (j=0; j<n; ++j)
```

```
if (i==j)
```

```
{
```

```
cout << A[i][j];
```

```
k = k + 1;
```

```
}
```

```
cout << "In k = " << k;
```

```
}
```

```
S = S + A[i][j]
```

ملاحظة: مجموع عناصر

القطر الرئيسي

المتباعد عن القطر الرئيسي بالعدد 25.

```
cout << A[i][j]; cout << 25;
```

س. التعامل مع المتغيرات الحرفية والسلاسل الحرفية.

المتغيرات الحرفية char

وتتضمن الأحرف الكبيرة والأحرف الصغيرة والرموز والإشارات وصيغتها:

char اسم متغير

char

مثال



عند إعطاء قيمة متغيرة من قبل، القيمة سيخبرنا أن المتغير تم تعريفه مسبقاً.  
مثال:

```
c = 'A';  
cout << c; A
```

بواسطة امر الطباعة. cout سيكتب طباعة الرمز أو طباعة رقم الترتيب.  
مثال:

```
int c;  
c = 'A';  
cout << c; 65  
char c;  
c = 'A';  
cout << c; A
```

تمرين (وظيفة):  
اكتب برنامج يطبع جداول الأسس كود  
مثال:

اكتب برنامج يطبع الأحرف الكبيرة A, B, ..., Z.  
الحل:

```
#include <iostream.h>  
void main()  
{  
    char i;  
    for (i = 'A'; i <= 'Z'; ++i)  
        cout << i;  
}
```

طباعة ارقام الترتيب نكتب بدل char i نكتب int i

تمرين (وظيفة):

اكتب برنامج يطبع الأحرف الكبيرة وارقام الترتيب.

تمرين: اكتب برنامج يسألنا عن ارقام الترتيب المطابق حرف حرف نكتب ارقام  
الأحرف بالانجليزي (A-Z) وطباعة عدد أحرف (A-Z) المطابق (A-Z) المطابق (A-Z).  
الحل:



```

#include <iostream.h>
void main()
{
    char c;
    int i=0;
    do
    {
        cin >> c;      طباعة
        cout << c;
        i = i+1;
    }
    while (c != '*');
    cout << "ini = " << i;
}

```

سند التعامل مع السلسلة الحرفية  
 ١- عبارة عن سلسلة أو تسلسل من الرموز أو الأحرف والأرقام والإشارات... الخ  
 نوع المتغيرات char يتقدم من أجل القابل مع المتغيرات الحرفية. إشارته إلى ذلك  
 نوع المتغيرات char يتقدم من أجل التعامل مع السلسلة الحرفية ولها قيم  
 التي تحد طول السلسلة الحرفية عند تعريفها، السلسلة الحرفية:

[طول السلسلة] اسم السلسلة char

مثال:

char s[15];

هنا لدينا سلسلة حرفية اسمها s مكونة من (15) حرف.  
 عند إعطاء قيمة للسلسلة الحرفية يجب عدم هذه القيمة يتعدى حجمها من دونه

مثال:

s = "Borland";

يمكن أيضاً تعريف السلسلة بإعطاءها قيمة:

مثال:

char s[6] = "basic";



عند التعامل مع السلسلة الحرفية يمكن في الواقع التعامل مع السلسلة الحرفية حرف حرف أو التعامل مع السلسلة الحرفية اسم.

مثال: `char s[6] = {'a', 'b', 'c', 'd', 'e', 'f'}`

السلسلة الحرفية (الهيكلية).

تتم إعادة كتابة السلسلة الحرفية الهيكلية في ذاكرة البرنامج، ولا يمكن الوصول إليها من قبل المترجم اللغة.

لذلك يجب أن نخزن موضع الحرف الهيكلية. ويمكن أيضًا تعريف السلسلة الحرفية كاسم أو سلسلة ويمكن عمل اسمين في اسماتي، تنقسم مزدوجة.

مثال: `char s[6] = "basic";`

في مثال السلسلة الحرفية حرف حرف، نستخدم حلقة `for` لإدخال السلسلة الحرفية كاسم من لوحة المفاتيح، نستخدم الأمر:

`cin >> s;`

اسم السلسلة.

مثال: !

`char st[20];`

`cin >> st;`

عند إدخال السلسلة من لوحة المفاتيح، نستخدم السلسلة اسم من دون وجود الفراغ، لأن الفراغ يملك فاصل بين السلسلة. وعند إدخال الاسم أو السلسلة الحرفية من لوحة المفاتيح، وجود الفراغ، نستخدم الأمر:

`gets(st);`

وهذا يقدم الأمر `gets` يقدم الأمر:

`#include <stdio.h>`

والطابعة السلسلة الحرفية نستخدم الأمر `cout`



cout << st ;

مثال ١

لطباعة سلسلة حرفية تحتاج إلى حلقة for

هناك بعض الدوال التي تستخدم مع المتغيرات الحرفية :

١- الدالة (تغير) toupper

تحويل الحرف الصغير إلى حرف كبير.

٢- الدالة to lower

تحويل الحرف الكبير إلى حرف صغير.

حيث نستخدم هذه الأسماء في البرامج.

#include <ctype.h>

#include <ctype.h>

مثال ٢

char c = 'd';

char A = 'G';

Lower(c); ١

to lower(A); ٩

ويوجد أيضاً الملف <string.h>

#include <string.h>

للتعامل مع السلسلة الحرفية وهناك بعض الدوال التي تستخدم مع السلسلة الحرفية وهي :

قيمة ←  
↑  
تغير  
Strcpy (s1, s2) ;

١

وتعني نسخ قيمة السلسلة الحرفية إلى متغير.

مثال ٣

char A[20];

char B[10] = "Berland";

يتم نسخ Berland إلى A (أي نسخ قيمة المتغير).

Strcomp (s1, s2) ;

٢

المقارنة بين السلسلتين وتتوقف عملية المقارنة عند أول حرفين غير متطابقين.



وتعطي قيمة موجبة إذا كان  $S_1 > S_2$   
 وقيمة سالبة إذا كان  $S_1 < S_2$   
 وقيمة صفرية إذا كان  $S_1 = S_2$

③ -  $strcat(S_1, S_2)$  ;

تقدم من أجل ربط الخاتمة السلسلة الثانية في نهاية السلسلة الأولى.  
 ويجب أن تكون السلسلة الأولى كبيرة بقدر كاف حتى تتسع للسلسلة.  
 مثال :

`char A(70) = "barland";`

`char B(10) = "c++";`

④ -  $strupper(st)$  تحويل سلسلة لأحرف كبيرة.

⑤ -  $strlower(st)$  تحويل سلسلة لأحرف صغيرة.

⑥ -  $strlen(s)$  ;

تقدم من أجل أن نعطينا طول السلسلة (عدد الأحرف).  
 وهذه الدالة تقدم مع الأمر (الملف الرئيسي).

`#include <string.h>`

تمرين : اكتب برنامج يعرف سلسلة حرفية ويعطى قيمة لها  
 ثم اطلع هذه السلسلة.  
 ① - طباعة السلسلة اسم كامل.  
 ② - طباعة السلسلة حرف حرف.

انتهى المحاضرة